

* SQL uses queries that retrieve, filter, and manipulate data using a variety of operators, functions and clauses. It covers essential aspects like Conditional Filtering and logical operators, pattern matching 'like' and 'not like', range checks using 'Between... And', and handling of null values. In addition, it delves into grouping and ordering data, applying aggregate functions, and refining results with the 'Having' clause. The use of Set operators to combine results sets, along with single row functions for handling date, time, and strings, is also discussed making SQL a versatile language for efficient data management.

1) Conditional and logical operators (where clause). Filters rows based on multiple conditions using AND, OR operators.

Syntax :-
 Select Column_name(s)
 From table_name
 Where Condition1 AND Condition2 OR Condition3;

2) LIKE / NOT LIKE:

Filters rows where a column's value matches or doesn't match a specific pattern

Syntax :-
 Select Column_name
 From table_name
 Where Column_name Like 'pattern';
 // OR
 Where Column_name NOT Like 'Pattern';

3) IN / NOT IN:

Filters rows where a Column's value is in or not in a specified list.

Syntax :- ~~IN~~ Select rows.

Select Column_name

From table_name

Where Column_name IN (value1, value2, ...);

or

For NOT IN

Select Column_name

From table_name

Where Column_name NOT IN (value1, value2, ...);

4) BETWEEN ... AND

Filters rows where a Column's value falls within a specified range

Syntax: Select Column_name

From table_name

Where Column_name BETWEEN value1 AND value2;

5) IS NULL / IS NOT NULL

Filters rows where Column has or doesn't have null values

Syntax: IS NULL :

Select Column_name

From table_name

Where Column_name IS NULL;

IS NOT NULL

Select Column_name

From table_name

Where Column_name IS NOT NULL;

6) ORDER BY:

Sort the result set by specified columns in ascending or descending order.

Syntax: SELECT column_name
FROM table_name
ORDER BY column_name ASC/DESC;

7) Group By

Group rows sharing a property and allows aggregate function to be applied

Syntax: SELECT column_name, COUNT(*)
FROM table_name
GROUP BY column_name;

8) Aggregate Functions (SUM, COUNT, AVG, MIN, MAX):
performs calculations on a set of rows and returns a single value.

Syntax:

SELECT SUM(column_name), COUNT(column_name), AVG(column_name),
MIN(column_name), MAX(column_name)
FROM table_name;

9) Having Clause (with Group By):

Filters groups based on aggregate function results, similar to where but for groups.

Syntax: SELECT column_name, COUNT(*)
FROM table_name
GROUP BY column_name
HAVING COUNT(*) > value;

(a) Set operators (Union, Intersection, Except):

Combines result sets from multiple queries, returning union, intersection, or differences.

Syntax:

- `SELECT column_name FROM table1`
`UNION`
`SELECT column_name FROM table2;`

- `SELECT column_name FROM table1`
`UNION INTERSECT`
`SELECT column_name FROM table2;`

- `SELECT column_name FROM table1`
`EXCEPT`
`SELECT column_name FROM table2;`

ii) Single-row functions (Date, Time, String):

- **Date / Time Functions:** Returns and manipulates data and time values.

Syntax:

`SELECT Current_Date, Current_Time, NOW(), Date_ADD`
`(date_column, INTERVAL 10 Day);`

- **String Functions:** Performs operations like Base Conversion and Substring Extraction on strings.

Syntax:

`SELECT UPPER(column_name), LOWER(column_name),`
`LENGTH(column_name), substring(column_name);`